UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR		ATTORNEY DOCKET NO.
09/512,592	02/23/00	DICKENS	B	2039-301

Z2853
LEVIN & HAWES
P O BOX 4140
LAGUNA BEACH CA 92652

TM31/0918

EXAMINER	
HOMERE, J	
ART UNIT	**PAPER NUMBER**
2177	

DATE MAILED: 09/18/01

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

Office Action Summary

Application No.	Applicant(s)
09/512,592	Dickens
Examiner	**Art Unit**
Jean R. Homere	2177

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *three* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *Jan 5, 2001*

2a) ☒ This action is **FINAL.** 2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1-76* is/are pending in the applica

4a) Of the above, claim(s) *none* is/are withdrawn from considera

5) ☐ Claim(s) ____ is/are allowed.

6) ☒ Claim(s) *1-76* is/are rejected.

7) ☐ Claim(s) ____ is/are objected to.

8) ☐ Claims ____ are subject to restriction and/or election requirem

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on ____ is/are objected to by the Examiner.

11) ☐ The proposed drawing correction filed on ____ is: a) ☐ approved b) ☐ disapproved.

12) ☒ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

13) ☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

a) ☐ All b) ☐ Some* c) ☐ None of:

1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ____
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

*See the attached detailed Office action for a list of the certified copies not received.

14) ☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

15) ☐ Notice of References Cited (PTO-892)

16) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

17) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s). *3, 12, 1*

18) ☐ Interview Summary (PTO-413) Paper No(s). ____

19) ☐ Notice of Informal Patent Application (PTO-152)

20) ☐ Other:

DETAILED ACTION

Reissue Applications

Objection to Oath/Declaration

1. The reissue oath/declaration filed with this application is defective because it fails to identify at least one error which is relied upon to support the reissue application. See 37 CFR 1.175(a)(1) and MPEP § 1414.

a. The declaration merely provides a statement of "inadequate claiming" as an error for the instant reissue. The "inadequate claiming" statement does not meet the requirement, as provided in MPEP § 1414.

b. A reissue declaration must be signed by the inventor when the claims are being broadened. Although Mr. Dickens signed the present declaration as the assignee's representative, the inventor's averments provided herein are deemed to be adequate to satisfy the inventor's signature requirement. Consequently, the inventor's declaration provided herein is hereby accepted as the reissue declaration.

c. The Oath/Declaration fails to identify the inventor's full name as well as his country of citizenship, as required by 35 U.S.C. 115 and 37 CFR 1.63(a)(3). 1.63

ASSENT OF ASSIGNEE UNACCEPTABLE

2. This application is objected to under 37 CFR 1.172(a) as the assignee has not established its ownership interest in the patent for which reissue is being requested. An assignee must establish its ownership interest *in order to support the consent to a reissue application as required by 37 CFR 1.172*(a). The submission establishing the ownership interest of the assignee is informal. There is no indication of record that the party who signed the submission is an appropriate party to sign on behalf of the assignee. 37 CFR 3.73(b). Further, the establishment of ownership under 37 CFR 3.73 is insufficient since it relies upon certain attached documents, which have not been supplied to the office. Consequently, a proper submission establishing ownership interest in the patent, pursuant to 37 CFR 1.172(a), is required in response to this action.

3. This application is objected to under 37 CFR 1.172(a) as lacking the written consent of all assignees owning an undivided interest in the patent. The consent of the assignee must be in compliance with 37 CFR 1.172. See MPEP § 1410.01. A proper assent of the assignee in compliance with 37 CFR 1.172 and 3.73 is required in reply to this Office action.

4. The original patent, or an affidavit or declaration as to loss or inaccessibility of the original patent, must be received before this reissue application can be allowed. See 37 CFR 1.178.

5. Changes made in the certificate of correction have not been incorporated into the specification of the reissue application. Applicant is required submit a substitute specification which complies with reissue practice.

6. Claims 1-76 are rejected as being based upon a defective reissue declaration under 35 U.S.C. 251 as set forth above. See 37 CFR 1.175. The nature of the defect(s) in the declaration is set forth in the discussion above in this Office action.

Suggestions and Recommendations

6a. Regarding paragraph 1, the applicant's attention is directed to MPEP 1414, stating that: " In identifying the error, it is sufficient that the reissue oath/declaration identify a single word, phrase, or expression in the specification or in an original claim, and how it renders the original patent wholly or partly inoperative or invalid."

6b. Regarding paragraphs 2-6, the applicant is advised to submit copies of the records evidencing the chain of assignment. The declaration should also be signed by the inventor acting as inventor in addition to acting on behalf of the assignee. Further, the statement under 37 CFR 3.73 should identify the relationship of the inventor to the assignee upon which the inventor relies to authorize his signature as one in which he is authorized to act on behalf of the assignee (i.e. what is his corporate office in the assignee.) Models of Consent by the assignee can be found on the USPTO web site at: http://www.uspto.gov/web/forms/sb0053.pdf

10. Claims 1-3, 5, 7, 9-10 are rejected under 35 U.S.C. 102(e) as anticipated by Daniel P. Shaughnessy, US. Patent No. 5, 630,118, filed on November 21, 1994 and issued on May 13, 1997 (Shaughnessy, hereinafter) or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Masakazu Hazama, Japanese Application No. 05-027947, published on February 5, 1993 (Hazama, hereinafter).

As to claim 1, Shaughnessy discloses the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date. Further, Shaughnessy discloses the claimed 'all of the symbolic representations of dates falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a 10-decade window with a Y.sub.A Y.sub.B value for the first decade of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1

Y.sub.2 year designator in the database' as a subroutine for determining the current date[1] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[2] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of ' reformatting the symbolic representation of the date with the values C.sub.1 C.sub.2, Y.sub.1 Y.sub.2, M.sub.1 M.sub.2 , and D.sub.1 D.sub.2 to facilitate further processing of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year

---

[1]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[2] Shaughnessy specifically suggests that it might be desirable to set the current date to a date w... compares low to all other dates (col. 7, lines 16-17 et seq).

period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

As to claim 2, Shaughnessy discloses the invention as discussed in the rejection of claim 1, as well as the claimed limitation whereby a '10-decade window includes the decade beginning in the year 2000' by suggesting the use of a 100 year window that includes a decade date in the 21st century (col. 6, lines 28-29 et seq).

As to claim 3, Shaughnessy discloses the invention as discussed in the rejection of claim 2, as well as the claimed limitation whereby, 'the step of determining includes the step of determining the first value as 20 and the second value as 19' by assigning the century value to 19 if the YYDDD portion of the date is greater than or equal to

corresponding portion of the corresponding portion of the modified system install date (col. 5, lines 40-46) and by assigning the century value to 20 if the pivot date is less than the modified system install date (col.5, lines 52-60 et seq).

As to claim 5, Shaughnessy discloses the invention as discussed in the rejection of claim 1, as well as the claimed limitation, wherein 'the step of reformatting includes the step of reformatting each symbolic representation of a date into the format C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2' as the conversion of the current date from a six digit format (YYMMDD) into an 8-digit format (CCYYMMDD) (col. 5, lines 48-50 et seq).

As to claim 7, Shaughnessy discloses the invention as discussed in the rejection of claim 1, as well as the claimed limitation, wherein the step of providing a database includes the step of converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by as the converting the current date in a six digit format (YYMMDD), wherein YY represents the year, MM represents the month and DD represents the day (col. 8, lines 18-27 et seq).

As to claim 9, Shaughnessy discloses the invention as discussed in the rejection of claim 1, as well as the claimed limitation of 'storing the symbolic representation of dates and their associated information back into the database after the step of reformatting' by saving the converted date in the database (col. 6, lines 1-3 et seq).

As to claim 10, Shaughnessy discloses the invention as discussed in the rejection of claim 9, as well as the claimed limitation of ' manipulating information in the database having the reformatted date information therein' by performing updates on the converted dates and saving said converted dates in the database (col. 6, lines 1-22 et seq).

11. Claims 4, 6, 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth et al., Implementation in Clipper 5A Developer's Guide (Booth, hereinafter) or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama, as applied to the rejection of claims 1-3, 5, 7, 9-10 above, further in view of Booth.

As to claim 4, Shaughnessy substantially discloses the invention as discussed in the rejection of claim 1 above. Shaughnessy does not, disclose, the additional step of 'sorting the symbolic representations of dates, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system

to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 6, Shaughnessy substantially discloses the invention as discussed in the rejection of claim 5 above. Shaughnessy does not specifically, disclose the additional step of 'sorting the symbolic representations of dates using a numerical-order sort, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 8, Shaughnessy substantially discloses the invention as discussed in the rejection of claim 1. Shaughnessy does not specifically, disclose the step of selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero). However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database

to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2 M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting that the pivot date be set to 90 by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of Shaughnessy's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

12. Claims 11-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 11, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method of processing dates in a database' ~~by providing~~ as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy further discloses the claimed step of 'providing a database with dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year

designator, all of dates falling within a 10-decade period of time which includes the decade beginning in the year 2000' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date, and wherein the ten decade includes a decade date in the 21st century (col. 6, lines 28-29 et seq). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a 10-decade window with a Y.sub.A Y.sub.B value for the first decade of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database' as a subroutine for determining the current date[3] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36).

Additionally, Shaughnessy discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD

[3]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, is therefore not any later than the earliest date in 100 year-cycle in the database.

portion[4] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of ' reformatting each date in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2 to facilitate further processing of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having

[4] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically, disclose the step of 'sorting the dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 12, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 11. Additionally, Shaughnessy discloses step of ' converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by converting the current date in a six digit format

(YYMMDD), wherein YY represents the year, MM represents the month and DD represents the day (col. 8, lines 18-27 et seq).

As to claim 13, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 11. Booth further complements Shaughnessy by disclos~~es~~ing the claimed ~~the~~ step of 'selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero) ' by suggesting that the pivot date be set to 90 and by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of Shaughnessy's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

As to claim 14, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 11. Booth further complements Shaughnessy by disclosing the step of 'storing the sorted dates and their associated information back into the database' by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would allow users of Shaughnessy's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

As to claim 15, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 14. Additionally, Booth discloses the step of 'manipulating information in the database having the reformatted date therein' by renaming and storing the sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

13. Claims 16-18, 20, 22, 24-25 are rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view Hazama.

As to claim 16, Shaughnessy discloses the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date. Further, Shaughnessy discloses the claimed 'all of the symbolic representations of dates falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100

YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a window with a Y.sub.A Y.sub.B value for a pivot date of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database' as a subroutine for determining the current date[5] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[6] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of ' reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the

---

[5]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[6] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates.' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

As to claim 17, Shaughnessy discloses the invention as discussed in the rejection of claim 16, as well as the claimed limitation whereby 'the window includes at least a portion of the decade beginning in the year 2000' by suggesting the use of a 100 year window that includes a decade date in the 21st century (col. 6, lines 28-29 et seq).

As to claim 18, Shaughnessy discloses the invention as discussed in the rejection of claim 17, as well as the claimed limitation whereby, 'the step of determining includes the step of determining the first value as 20 and the second value as 19' by assigning the century value to 19 if the YYDDD portion of the date is greater than or equal to the corresponding portion of the corresponding portion of the modified system install date (col. 5, lines 40-46) and by assigning the century value to 20 if the pivot date is less than the modified system install date (col.5, lines 52-60 et seq).

As to claim 20, Shaughnessy discloses the invention as discussed in the rejection of claim 16, as well as the claimed limitation, wherein 'the step of reformatting includes the step of reformatting each symbolic representation of a date into the format C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2 separately from the symbolic representations in the database' as the conversion of the current date from a six digit format (YYMMDD) into an 8-digit format (CCYYMMDD) (col. 5, lines 48-50 et seq).

As to claim 22, Shaughnessy discloses the invention as discussed in the rejection of claim 16, as well as the claimed limitation, wherein the step of providing a database includes the step of converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1

D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by as the converting the current date in a six digit format (YYMMDD), wherein YY represents the year, MM represents the month and DD represents the day (col. 8, lines 18-27 et seq).

As to claim 24, Shaughnessy discloses the invention as discussed in the rejection of claim 16, as well as the claimed limitation of 'storing the symbolic representation of dates and their associated information back into the database after the step of reformatting' by saving the converted date in the database (col. 6, lines 1-3 et seq).

As to claim 25, Shaughnessy discloses the invention as discussed in the rejection of claim 24, as well as the claimed limitation of ' manipulating information in the database having the reformatted date information therein' by performing updates on the converted dates and saving said converted dates in the database (col. 6, lines 1-22 et seq).

14. Claims 19, 21, 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama, as applied to the rejection of claims 16-18, 20, 22, 24-25 above, further in view of Booth.

As to claim 19, Shaughnessy substantially discloses the invention as discussed in the rejection of claim 16 above. Shaughnessy does not, disclose, the additional step of 'sorting the symbolic representations of dates, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq). In particular,

analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 21, Shaughnessy substantially discloses the invention as discussed in the rejection of claim 20 above. Shaughnessy does not specifically, disclose the additional step of 'sorting the symbolic representations of dates using a numerical-order sort, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of

the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 23, Shaughnessy substantially discloses the invention as discussed in the rejection of claim 16. Shaughnessy does not specifically, disclose the step of selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero). However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting that the pivot date be set to 90 by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of Shaughnessy's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

15. Claims 26-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 26, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date. Further, Shaughnessy discloses the claimed 'all of the symbolic representations of dates falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy also discloses the claimed step of 'selecting a window with a Y.sub.A Y.sub.B value for a pivot date of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database' as a subroutine for determining the current date[7] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the

[7]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[8] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of ' reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates.' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

---

[8] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically, disclose the step of 'sorting the dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of

the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 27, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 26. Additionally, Shaughnessy discloses step of 'converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by converting the current date in a six digit format (YYMMDD), wherein YY represents the year, MM represents the month and DD represents the day (col. 8, lines 18-27 et seq).

As to claim 28, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 26. Booth further complements Shaughnessy by discloses the claimed ' the step of selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero) ' by suggesting that the pivot date be set to 90 and by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of Shaughnessy's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

As to claim 29, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 26. Booth further complements Shaughnessy by disclosing the step of 'storing the sorted dates and their associated information

back into the database' by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

As to claim 30, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 29. Additionally, Booth discloses the step of 'manipulating information in the database having the reformatted date therein' by renaming and storing the sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

16. Claim 31 is rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy, or in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 31, Shaughnessy discloses the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with symbolic representations of

dates stored therein according to a format wherein Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a window with a Y.sub.A Y.sub.B value for a pivot date of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database' as a subroutine for determining the current date[9] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[10] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the

---

[9] The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[10] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which

century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates.' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's. Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in

---

compares low to all other dates (col. 7, lines 16-17 et seq).

the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

17. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 32, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a window with a Y.sub.A Y.sub.B value for a pivot date of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database' as a subroutine for determining the current date[11]

[11]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the

to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[12] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each of the dates in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

---

operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[12] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically, disclose the step of ' sorting the dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date

[redacted]

into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

18. Claim 33 is rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 33, Shaughnessy discloses the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the

cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database' as a subroutine for determining the current date[13] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than A YB and having a second value if Y1 Y2 is equal to or greater than A YB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[14] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database without changing any of the symbolic representations of a date in the database during the reformatting step, with the reformatted symbolic representation of each date in the database having the values C1C2, Y1Y2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed

---

[13] The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[14] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

without changing any of the symbolic representations of a date in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

19. Claims 34-59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 34, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the step of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program collectively on each of the converted symbolic representations of each of the respective dates to manipulate the dates represented by the converted symbolic representations, separately from the date data symbolic representations contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for

determining the current date[15] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[16] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

[15]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[16] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically, disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of

the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 35, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 34. Additionally, Shaughnessy discloses step of 'opening the database prior to the step of converting' by providing a subroutine to retrieve a six digit date from its storage location in an existing application program (i.e. requires opening the DB, first) prior to converting said date to an eight digit format (col. 4, lines 29-33 et seq).

As to claim 36, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 37, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 38, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 39, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 40, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 41, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 42, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 43, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations according to a different data entry field contained in the database from the at least one date field,

prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 44, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Shaughnessy by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 45, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Shaughnessy by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 46, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 34. Shaughnessy further discloses the step of 'converting at least a substantial portion of each of the plurality of symbolic representations of dates in the at least one date field and repeating this step until each of the date data entries in the at least one date field is converted into the format that does not have the ambiguity' by converting the current date stored in the database field from an ambiguous six digit format (YYMMDD) into an unambiguous 8-digit format (CCYYMMDD), wherein the century for the date is specified (col. 5, lines 48-50 et seq).

As to claim 47, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 35. Shaughnessy further discloses the step of 'converting at least a substantial portion of each of the plurality of symbolic representations of dates in the at least one date field and repeating this step until each of the date data entries in the at least one date field is converted into the format that does not have the ambiguity' by converting the current date stored in the database field from an ambiguous six digit format (YYMMDD) into an unambiguous 8-digit format (CCYYMMDD), wherein the century for the date is specified (col. 5, lines 48-50 et seq).

As to claim 48, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 49, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 47. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 50, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 51, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 49. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 52, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field in the database than the at least one date field, prior to the step of running the program' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 53, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 47. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field in the database than the at least one date field, prior to the step of running the program' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 54, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 52. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 55, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 53. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 56, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 52. Booth further complements Shaughnessy by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 57, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 53. Booth further complements Shaughnessy by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 58, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 54. Booth further complements Shaughnessy by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 59, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 55. Booth further complements Shaughnessy by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

20. Claim 60 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 60, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates

ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program on each of the converted symbolic representations of each of the respective dates to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the date data symbolic representations of dates contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[17] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system

[17]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[18] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having

[18] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically, disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

21. Claim 61 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 61, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic

representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program collectively on each of the converted symbolic representations of each of the respective dates to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[19] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36).

[19]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[20] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the

[20] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically, disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

22. Claim 62 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 62, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program on the stored converted symbolic representations to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the symbolic representations of-dates--contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[21]

[21]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[22] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the

[22] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically, disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Shaughnessy does not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Shaughnessy by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to

one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

23. Claim 63 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 63, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program collectively on the stored converted symbolic representations to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates

contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[23] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[24] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that applicant were to argue that Shaughnessy does not disclose the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period,

---

[23]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[24] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically, disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH

command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945. Shaughnessy does not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Shaughnessy by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

24. Claim 64 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 64, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without

mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program on the stored converted symbolic representations to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[25] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively,

[25]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

Shaughnessy provides a subroutine that compares the YYMMDD portion[26] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having

[26] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically, disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq). In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Shaughnessy does not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field in the database.' Booth, however, further complements Shaughnessy by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

25. Claim 65 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 65, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the steps of ' converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program collectively on the stored converted symbolic representations to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a

particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[27] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[28] of the date to the corresponding date portion at the end of the 100 year-cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

---

[27]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[28] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of

the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945. Shaughnessy does not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field in the database.' Booth, however, further complements Shaughnessy by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

26. Claim 66 is rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 66, Shaughnessy discloses the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with dates stored in at least one date field therein according to a format wherein M1 M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day

and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a window with a Y.sub.A Y.sub.B value for a pivot date of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database' as a subroutine for determining the current date[29] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[30] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the

[29] The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[30] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1C2, Y1Y2, M1M2, and D1D2; and repeating the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that applicant were to argue that Shaughnessy does not disclose the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having

all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

27. Claim 67 is rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy, or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 67, Shaughnessy discloses the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with dates stored in at least one date field therein according to a format wherein Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a window with YAYB value for a pivot date of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database' as a

subroutine for determining the current date[31] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[32] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2; and repeating the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

---

[31]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[32] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

28. Claim 68 is rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy, or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 68, Shaughnessy discloses the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date

fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored in at least one date field therein according to a format wherein Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the at least one date field of the database' as a subroutine for determining the current date[33] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD

[33]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

portion[34] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates, by running a program on the reformatted symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that applicant were to argue that Shaughnessy does not disclose the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last

[34] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

29. Claim 69 is rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 69, Shaughnessy discloses the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with dates stored in at least one date field therein according to a format wherein Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this

limitation. Shaughnessy also discloses the claimed step of 'selecting a window with a YAYB value for a pivot year of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database' as a subroutine for determining the current date[35] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each date in the at least one date field of the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[36] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2;
sorting the reformatted symbolic representations of the dates in the form C1 C2 Y1 Y2; and running a program on the reformatted symbolic representations of each of the dates' by appending the determined century value before the

---

[35]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[36] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

30. Claim 70 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 70, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year, with the pivot year being less than or equal to the earliest date represented by the symbolic representation of dates stored in the at least one date field, without the addition of any new data field to the database, and without modifying any of the symbolic representations of dates in the at least one date field, for purposes of such windowing and converting; and running a program on the converted symbolic representations of each of the dates to manipulate the dates represented by the converted symbolic representations, separately from the date data symbolic representations contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE

CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[37] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[38] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's. Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit

[37]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[38] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

31. Claim 71 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 71, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method for representing and utilizing dates stored in at least one date field of the database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year, with the pivot year being less than or equal to the earliest date represented by a symbolic representation of dates stored in the at least one date field, and without the addition of any new data field to the database for purposes of such windowing and converting; and running a program on the stored converted symbolic representations of each of the converted symbolic representations of the dates to manipulate the dates represented by the converted symbolic representations, separately from the date data symbolic representations contained in the at least one date field of the database. ' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day

and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[39] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[40] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

---

[39]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[40] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of

the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Shaughnessy does not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field in the database.' Booth, however, further complements Shaughnessy by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

32. Claims 72 is rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 72, Shaughnessy discloses the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col. 1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'selecting a database with symbolic representations of dates stored therein according to a format wherein Ml M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date

type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of ' selecting a 10-decade window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1Y2 year designator in the database' as a subroutine for determining the current date[41] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of ' determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[42] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq).

---

[41] The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[42] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database with the values C1 C2, Y1 Y2, M1 M2, and Dl D2 prior to collectively further processing information contained within the database associated with the respective dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

33. Claims 73 is rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy, or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 73, Shaughnessy discloses the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of ' providing a database with symbolic representations of dates stored therein according to a format wherein Y1Y2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a 10-decade window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1Y2 year designator in the database' as a subroutine for determining the current date[43] to thereby select a 100 year cycle wherein the current date is the pivot

---

[43]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the

date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[44] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of the date with the values C1 C2, Y1 Y2, to facilitate further processing of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that applicant were to argue that Shaughnessy does not disclose the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

---

operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[44] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

34. Claim 74 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 74, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1 Y2 is the numerical year designator, all of symbolic representations of dates falling within a 10-decade period of time' as a database

having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a 10-decade window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1Y2 year designator in the database' as a subroutine for determining the current date[45] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[46] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq).

---

[45]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[46] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

Finally, Shaughnessy discloses the step of 'reformatting each date in the form C1C2Y1Y2 to facilitate further processing of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that applicant were to argue that Shaughnessy does not disclose the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper

programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

35. Claims 75 is rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy or, in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 75, Shaughnessy discloses the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M1M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD

represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS).' As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the YAYB earliest Y1 Y2 year designator in the database' as a subroutine for determining the current date[47] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[48] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the

[47] The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[48] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1C2, Y1Y2, MlM2, and D1 D2 in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

36. Claims 76 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 76, Shaughnessy substantially discloses the claimed invention. In particular, Shaughnessy discloses the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of ' providing a database with dates stored therein according to a format wherein Ml M2 is the numerical month designator, Dl D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of ' selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the earliest Y1Y2 year designator in the database; determining a century designator Cl C2 for each date in the database, Cl C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as a subroutine for determining the

current date[49] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[50] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values Cl C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year

---

[49]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[50] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

period, it would been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy does not specifically disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq). In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH

command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

37. Claims 1-3, 5, 7, 9-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over B.G. Ohms, *Computer processing of Dates Outside the Twentieth Century*, IBM Systems Journal, Volume 25, Number 2, 1986, pages 244-251, (Ohms, hereinafter), in view of Hazama.

As to claim 1, Ohms substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). In particular, Ohms discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq). Ohms also discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' by indicating that years that are later or equal (25-99) to the

pivot date (25) would fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date would fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Finally, OHMS discloses the step of 'reformatting the symbolic representation of the date with the values C.sub.1 C.sub.2, Y.sub.1 Y.sub.2, M.sub.1 M.sub.2 , and D.sub.1 D.sub.2 to facilitate further processing of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century[51], it is expressed in accordance with its corresponding century (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a 10-decade window with a Y.sub.A Y.sub.B value for the first decade of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This

[51]Ohms implicitly discloses that C1C2 corresponds to 19 or 20 depending on whether the date is less than or greater than or equal to the pivot date.

determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

As to claim 2, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses that 'the 10-decade window includes the decade beginning in the year 2000' by indicating that the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column).

As to claim 3, Ohms and Hazama disclose the invention as discussed in the rejection of claim 2. Additionally, Ohms discloses that the step of 'determining includes the step of determining the first value as 20 and the second value as 19' by indicating that dates that are greater or equal to the pivot date fall within the 20th century (C1C2=19) and dates that are less than the pivot date fall within the 21st century (C1C2=20) p. 2488, right hand column).

As to claim 5, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses that the step of 'reformatting includes the step of reformatting each symbolic representation of a date into the format C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2 by indicating that dates that fall within the 20th century (greater than or equal to the pivot date) are preceded by 19 (e.g. 1925-1999), whereas dates that fall within the 21st century (less than the pivot date) are preceded by 20 (e.g. 2000-2024) p 2477, right hand column).

As to claim 7, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses that the step of 'providing a database includes the step of converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by suggesting the conversion from of a date from a Gregorian format (MMDDYYYY) to a short Gregorian format (YYMMDD), wherein YY indicates the year, MM indicates the month and DD indicates the date p 2477, table 1).

As to claim 9, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses, 'after the step of reformatting, the storing of the symbolic representation of dates and their associated information back into the database' by suggesting that converted eight dates are stored in the database although they take up more memory space than non-converted six digits dates p 2499, left hand column).

As to claim 10, Ohms and Hazama disclose the invention as discussed in the rejection of claim 9. Additionally, Ohms discloses, 'after the step of reformatting, the manipulating of information in the database having the reformatted date information therein' by suggesting that the converted dates can be saved in the database p 2499, left hand column).

38. Claims 4, 6, 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, as applied to the rejection of claims 1-3, 5, 7, 9-10 above, further in view of Booth.

As to claim 4, OHMS and Hazama substantially disclose the invention as discussed in the rejection of claim 1 above. Ohms and Hazama do not particularly disclose the additional step of 'sorting the symbolic representations of dates, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper

programming language to process dates stored in a database to thereby derive other dates therefrom p 939, lines 1-3 et seq. In particular, analogously to Ohms, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into p 9411, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 6, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 5 above. Ohms and Hazama do not specifically, disclose the additional step of 'sorting the symbolic representations of dates using a numerical-order sort, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of

the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 8, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 1. Ohms and Hazama do not specifically, disclose the step of selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero). However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting that the pivot date be set to 90 by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Ohms-Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

39. Claims 11-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 11, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator, all of dates falling within a 10-decade period of time which includes the decade beginning in the year 2000' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, OHMS discloses the step of 'reformatting each date in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2 to facilitate further processing of the dates' by indicating that upon determining that a two-digit date

falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a 10-decade window with a Y.sub.A Y.sub.B value for the first decade of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of ' sorting the dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates

therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 12, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 11. Additionally, Ohms discloses step of ' converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator' by suggesting the conversion from of a date from a Gregorian format (MMDDYYYY) to a short Gregorian format (YYMMDD), wherein YY indicates the year, MM indicates the month and DD indicates the date (p 247, table 1).

As to claim 13, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 11. Booth further complements Ohms and Hazama by disclosing the claimed ' the step of selecting Y.sub.A Y.sub.B such that

Y.sub.B is 0 (zero) ' by suggesting that the pivot date be set to 90 and by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Ohms-Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

As to claim 14, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 11. Booth further complements Ohms and Hazama by disclosing the step of 'storing the sorted dates and their associated information back into the database' by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

As to claim 15, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 14. Additionally, Booth discloses the step of 'manipulating information in the database having the reformatted date therein' by renaming and storing the sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Ohms-Hazama's system

to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

40. Claims 16-18, 20, 22, 24-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama. As to claim 16, Ohms substantially discloses the invention including the claimed ‘method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). In particular, Ohms discloses the step of ‘providing a database with symbolic representations of dates stored therein according to a format wherein M1, M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq). Ohms also discloses the step of ‘determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) would fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date would fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Finally, Ohms discloses the step of ‘reformatting the symbolic representation of each symbolic representation of a date in the database, without the

addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century[52], it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column). Regarding the step of 'selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the earliest Y1Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the

[52]Ohms implicitly discloses that C1C2 corresponds to 19 or 20 depending on whether the date is less than or greater than or equal to the pivot date.

dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

As to claim 17, Ohms and Hazama disclose the invention as discussed in the rejection of claim 16. Additionally, Ohms discloses that 'the 10-decade window includes at least a portion of the decade beginning in the year 2000' by indicating that the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column).

As to claim 18, Ohms and Hazama disclose the invention as discussed in the rejection of claim 17. Additionally, Ohms discloses that the step of ' determining includes the step of determining the first value as 20 and the second value as 19' by indicating that dates that are greater or equal to the pivot date fall within the 20th century (C1C2=19) and dates that are less than the pivot date fall within the 21st century (C1C2=20) (p 248, right hand column).

As to claim 20, Ohms and Hazama disclose the invention as discussed in the rejection of claim 16. Additionally, Ohms discloses that the step of 'reformatting includes the step of 'reformatting each symbolic representation of a date into the format C1 C2 Y1 Y2 Ml M2 Dl D2 separately from the symbolic representations in the database' by indicating that dates that fall within the 20th century (greater than or equal to the pivot date) are preceded by 19

(e.g. 1925-1999), whereas dates that fall within the 21st century (less than the pivot date) are preceded by 20 (e.g. 2000-2024) (p 247, right hand column).

As to claim 22, Ohms and Hazama disclose the invention as discussed in the rejection of claim 16. Additionally, Ohms discloses that the step of 'providing a database includes the step of converting pre-existing date information having a different format into the format wherein M1 M2 is the numerical month designator, D1 D2 is the numerical day designator and Y1 Y2 is the numerical year designator' by suggesting the conversion from of a date from a Gregorian format (MMDDYYYY) to a short Gregorian format (YYMMDD), wherein YY indicates the year, MM indicates the month and DD indicates the date (p 247, table 1).

As to claim 24, Ohms and Hazama disclose the invention as discussed in the rejection of claim 16. Additionally, Ohms discloses, 'after the step of reformatting, the storing the symbolic representation of dates and their associated information back into the database' by suggesting that converted eight dates are stored in the database although they take up more memory space than non-converted six digits dates (p 249, left hand column).

As to claim 25, Ohms and Hazama disclose the invention as discussed in the rejection of claim 24. Additionally, Ohms discloses, 'after the step of reformatting, the manipulating of information in the database having the reformatted date information therein' by suggesting that the converted dates can be saved in the database (p 249, left hand column).

41. Claims 19, 21, 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, as applied to the rejection of claims 16-18, 20, 22, 24-25 above, further in view of Booth.

As to claim 19, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 16 above. Ohms and Hazama do not particularly disclose the additional step of 'sorting the symbolic representations of dates, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 21, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 16 above. Ohms and Hazama do not specifically, disclose the additional step of 'sorting the symbolic representations of dates using a numerical-order sort, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY)

into A corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 23, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 16. Ohms and Hazama do not specifically, disclose the step of 'selecting includes the step of: selecting YAYB such that YB is 0 (zero).' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting that the pivot date be set to 90 by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Ohms-Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

42. Claims 26-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 26, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with dates stored therein according to a format wherein M1 M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the

database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1C2Y1Y2 M1M2, and D1 D2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two-digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of 'sorting the dates in the form C1C2Y1Y2M1 M2 Dl D2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 27, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 26. Additionally, Ohms discloses step of ' converting pre-existing date information having a different format into the format wherein M1 M2 is the numerical month designator, D1 D2 is the numerical day designator and Y1 Y2 is the numerical year designator' by suggesting the conversion from of a date from a Gregorian format (MMDDYYYY) to a short Gregorian format (YYMMDD), wherein YY indicates the year, MM indicates the month and DD indicates the date (p 247, table 1).

As to claim 28, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 26. Booth further complements Ohms and Hazama by disclosing the claimed the step of 'selecting includes the step of selecting YAYB such that YB is 0 (zero)' by suggesting that the pivot date be set to 90 and by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Ohms-Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

As to claim 29, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 26. Booth further complements Ohms and Hazama by disclosing, after the step of sorting, the step of 'storing the sorted dates and their associated information back into the database' by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

As to claim 30, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 29. Additionally, Booth discloses the step of ' manipulating information in the database having the reformatted dates therein' by renaming and storing the sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It

would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

43. Claim 31 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms, in view of Hazama.

As to claim 31, Ohms substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). In particular, Ohms discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or, greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) would fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date would fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column.

Finally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century[53], it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting A window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in

---

[53] Ohms implicitly discloses that C1C2 corresponds to 19 or 20 depending on whether the date is less than or greater than or equal to the pivot date.

symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for a pivot year of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH

paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

45. Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama.

As to claim 33, Ohms substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' by presenting A computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). In particular, Ohms discloses the step of 'providing A database with symbolic representations of dates stored therein according to a format wherein Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) would fall

within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date would fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column). Finally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database without changing any of the symbolic representations of a date in the database during the reformatting step, with the reformatted symbolic representation of each date in the database having the values C1C2Y1Y2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century[54], it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data

---

[54]Ohms implicitly discloses that C1C2 corresponds to 19 or 20 depending on whether the date is less than or greater than or equal to the pivot date.

processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

46. Claims 34-59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 34, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program collectively on each of the converted symbolic representations of each of the respective dates to manipulate the dates represented by the converted symbolic representations, separately from the

date data symbolic representations contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding A new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in

the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 35, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Additionally, Ohms discloses step of 'opening the database prior to the step of converting' by providing a subroutine to retrieve a six digit date from its storage location in an existing application program (i.e. requires opening the DB, first) prior to converting said date to an eight digit format (p 248, right hand column et seq).

As to claim 36, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 37, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 38, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 39, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted

symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 40, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 41, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 42, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations according to A different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that A string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in A field of the database, are sorted accordingly in A different field of the database (page 839-40 et seq).

As to claim 43, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations according to a different data entry field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 44, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted

dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 45, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 46, Ohms, Hazama, and Booth discloses the invention as discussed in the rejection of claim 34. Ohms further discloses the step of 'converting at least a substantial portion of each of the plurality of symbolic representations of dates in the at least one date field and repeating this step until each of the date data entries in the at least one date field is converted into the format that does not have the ambiguity' by converting the current date stored in the database field from an ambiguous six digit format (YYMMDD) into an unambiguous 8-digit format (CCYYMMDD), wherein the century for the date is specified (p 248, right hand column et seq).

As to claim 47, Ohms, Hazama and Booth discloses the invention as discussed in the rejection of claim 35. Ohms further discloses the step of 'converting at least a substantial portion of each of the plurality of symbolic representations of dates in the at least one date field and repeating this step until each of the date data entries in the

at least one date field is converted into the format that does not have the ambiguity' by converting the current date stored in the database field from an ambiguous six digit format (YYMMDD) into an unambiguous 8-digit format (CCYYMMDD), wherein the century for the date is specified (p 248, right hand column et seq).

As to claim 48, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 49, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 47. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that A string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 50, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted

symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 51, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 49. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 52, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field in the database than the at least one date field, prior to the step of running the program' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 53, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 47. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field in the database than the at least one date field, prior to the step of running the program' by suggesting that a string representation can be used to sort and index the converted dates

such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 54, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 52. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 55, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 53. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 56, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 52. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted

dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in A field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 57, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 53. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 58, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 54. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 59, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 55. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic

representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

47. Claim 60 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 60, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of A database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; running a program on each of the converted symbolic representations of each of the respective dates to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the date data symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates

(p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of A pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

48. Claim 61 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 61, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of

'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program collectively on each of the converted symbolic representations of each of the respective dates to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or A 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however,

discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's

system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

49. Claim 62 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 62, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program on the stored converted symbolic representations to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or A 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the

21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously

to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

50. Claim 63 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth..

As to claim 63, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic

representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program collectively on the stored converted symbolic representations to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's

system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

51. Claim 64 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 64, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of

the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program on the stored converted symbolic representations to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data

processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have

been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

52. Claim 65 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 65, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program collectively on the stored converted symbolic representations to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within

a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

53. Claim 66 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 66, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with dates stored in at least one date field therein according to a format wherein M1 M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having A first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, M1 M2, and D1 D2' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually

adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of ' selecting a window with YAYB value for A pivot date of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of 'repeating the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see

SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

54. Claim 67 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 67, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of ' providing a database with dates stored in at least one date field therein according to A format wherein Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having A first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e.

1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the earliest Y1Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in

the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of 'repeating the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

55. Claim 68 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama.

As to claim 68, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented

method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with symbolic representations of dates stored in at least one date field therein according to a format wherein Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier (00-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates, by running a program on the reformatted symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the at least one date field of the database,' Ohms discloses

specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

56. Claim 69 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 69, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with dates stored in at least one date field therein according to a format wherein Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall

within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the at least one date field of the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the at least one date field of the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window

is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of 'sorting the reformatted symbolic representations of the dates in the form C1 C2 Y1 Y2; and running a program on the reformatted symbolic representations of each of the dates' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

57. Claim 70 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 70, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program collectively on the stored converted symbolic representations to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding

century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date

(C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

58. Claim 71 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 71, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of the database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for

processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year, with the pivot year being less than or equal to the earliest date represented by a symbolic representation of dates stored in the at least one date field, and without the addition of any new data field to the database for purposes of such windowing and converting; and, running a program on the stored converted symbolic representations of each of the converted symbolic representations of the dates to sort or otherwise manipulate the dates represented by the converted symbolic representations, separately from the date data symbolic representations contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier (00-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding A new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have

been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

59. Claim 72 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama.

As to claim 72, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'selecting a database with symbolic representations of dates stored therein according to a format wherein Ml M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year

window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database with the values C1 C2, Y1 Y2, M1 M2, and Dl D2 prior to collectively further processing information contained within the database associated with the respective dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a 10-decade window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the

art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

60. Claim 73 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama.

As to claim 73, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1 Y2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than a and having a second value if Y1 Y2 equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall

within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of the date with the values C1 C2, Y1 Y2, to facilitate further processing of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a 10-decade window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

61. Claim 74 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 74, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1Y2 is the numerical year designator, all of symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date, fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting each date in the form C1 C2 Y1 Y2 to facilitate further processing of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of ‘selecting a 10-decade window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,’ Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form Cl C2 Y1 Y2.’ However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in A database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the

teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

62. Claim 75 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 75, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M1M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the

database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1C2, Y1Y2, MlM2, and D1 D2 in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the YAYB earliest Y1Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

63. Claim 76 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 76, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with dates stored therein according to a format wherein Ml M2 is the numerical month designator, Dl D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator Cl CZ for each date in the database, Cl C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values Cl C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data

field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of ' selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of 'sorting the dates in the form C1 C2 Y1Y2 M1M2 D1D2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by

suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jean R. Homere whose telephone number is (703)-308-6647. The examiner can normally be reached on Monday-Friday from 09:30 a.m.-6:00 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene, can be reached on Monday-Friday from 8:00 a.m. to 3:30 p.m. at (703)-305-9790.

Any response to this action should be mailed to: Commissioner of Patents and Trademarks Washington, D.C. 20231, **or faxed to:** (703) 746-7239, (for formal communications intended for entry), **or faxed to:** (703) 746-7238, (for after final communications intended for entry), **Or:** (703) 746-7240(for informal or draft communications, please label "PROPOSED" or "DRAFT"). Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA., Sixth Floor (Receptionist).

JEAN R. HOMERE
PRIMARY EXAMINER
9/17/01